

## Application Note: Image Timestamp on Prosilica GigE Series Cameras

By Arlin Kalenchuk, Applications Engineer at Allied Vision Technologies Canada Inc.

Each frame returned from the camera contains a timestamp (see PvApi.h):

`tPvFrame->TimestampHi`, `tPvFrame->TimestampLo`

This timestamp is assigned at the start of exposure for the corresponding frame. The number is in clock ticks, and counts up continuously from the start of camera booting. `TimeStampFrequency`, the clock rate in Hz, is a read only parameter on the camera.

Real world time =  $\text{TimeStamp} / \text{TimeStampFrequency}$

The camera clock crystal, a CITIZEN CS10-25.000MABJTR has a drift specification of +/- 50 PPM.

Users interested in synching the camera Timestamp to another clock should note that the camera clock will drift from the other clock, and therefore will need to be periodically re-synched by resetting the Timestamp.

**For PvAPI >= 1.22, see the `TimeStampReset`, `TimeStampValueLatch`, `TimeStampValueHi` and `TimeStampValueLo` attributes.**

For PvAPI < 1.22, or users developing their own driver and wishing to use direct register access:

To RESET the timestamp:

---

Timestamp Control Register address: `0x0944`, length 4 bytes. Bit 31 = reset. WRITE only. Do not READ this register.

To READ the timestamp:

---

Timestamp Control Register address: `0x0944`, length 4 bytes. Bit 30 = Latch current timestamp counter into Timestamp Value register. WRITE only. Do not READ this register.

Timestamp Value Register address: High part: `0x0948`, length 4 bytes. Low part: `0x094C`, length 4 bytes. It is necessary to latch the 64-bit timestamp value to guarantee its integrity when performing the two 32-bit read accesses. Access high first, then low. READ only.

---

Example code (See PvRegIo.h and examples/siotest for more on PvRegisterRead/Write):

---

```
unsigned long control_address = 0x0944;
// NOTE: Bit order is reversed:  [0 ----- 31]
unsigned long reset_data = 0x1;
unsigned long latch_data = 0x1 << 1;
unsigned long ts_address[2] = {0x0948, 0x094C};
unsigned long ts_data[2];
unsigned long num_complete;
tPvErr return_value;

// reset
return_value = PvRegisterWrite(handle, 1, &control_address, &reset_data, &num_complete);

// latch
return_value = PvRegisterWrite(handle, 1, &control_address, &latch_data, &num_complete);

// read timestamp
return_value = PvRegisterRead(handle, 2, &ts_address[0], &ts_data[0], &num_complete);
```